Class 05: Data viz with ggplot

Isabella Ruud

Graphs and plots in R

> Q. Which plot types are typically NOT used to compare distributions of numeric variables? **Network graphs**
>
> Q. Which statement about data visualization with ggplot2 is incorrect? **ggplot2 is the only way to create plots in R**

R has tons of different graphic systems. These include **"base R"** (e.g. the `plot()` function) and add on packages like **ggplot2**.

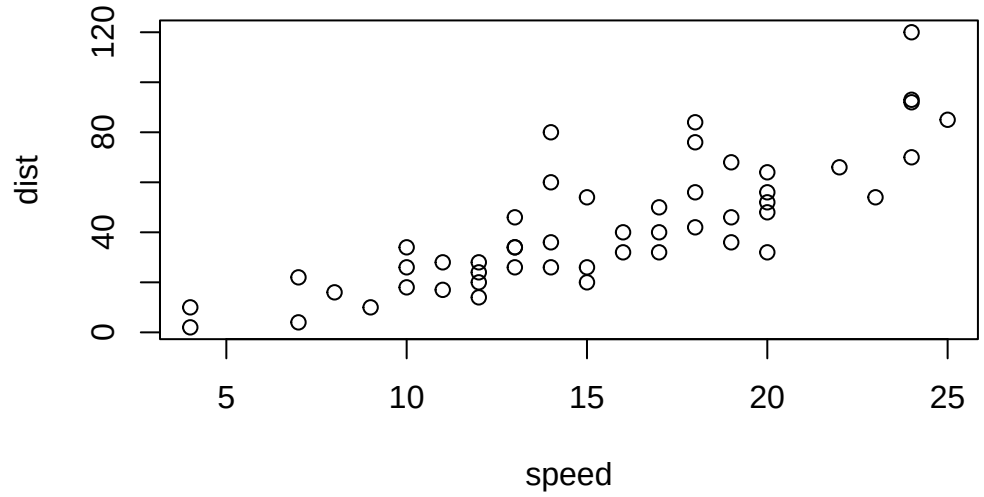
Let's start with plotting a simple dataset in "base R" and then ggplot2 to see how they differ.

```
head(cars)
```

```
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
```

To plot this in base R, I just use `plot()`.

```
plot(cars)
```



First to use ggplot2, I need to install the package. For this I use the `install.packages()` function.

> I don't want to run `install.packages()` in my quarto document as this would re-install the package every time I render the document.

The main function in the ggplot2 package is `ggplot()`. Before I can use this function, I need to load the package with a `library()` call.

```
library(ggplot2)
ggplot(cars)
```

There are 3 things that every ggplot needs:

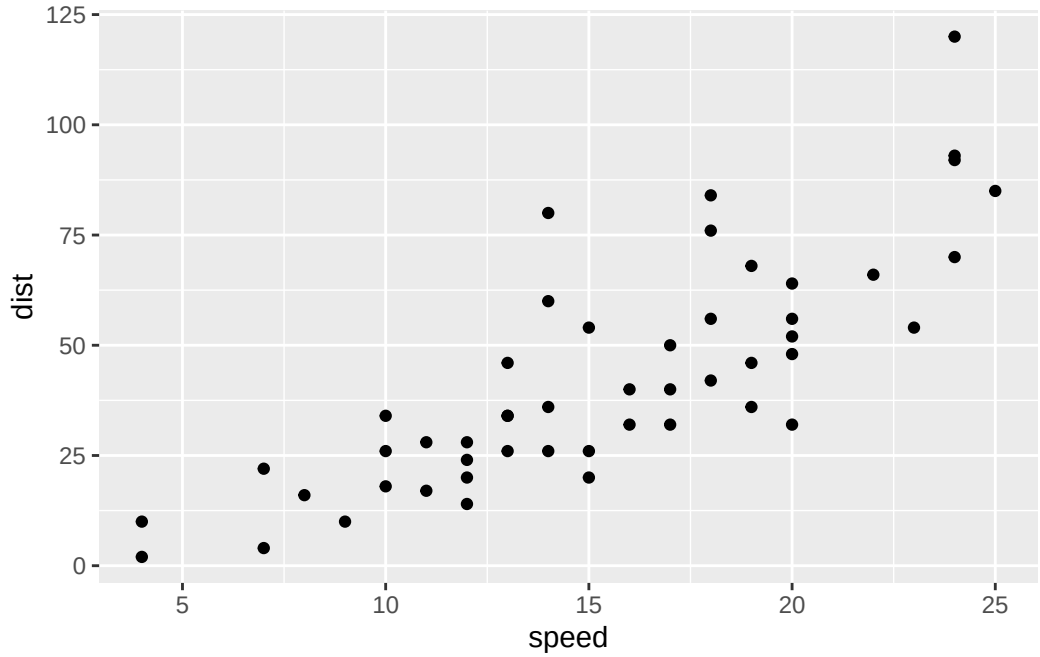
-the **data** (the data I want to plot)

-the **aes**thetics (how the data maps to my plot)

-the **geoms** or geometries (the style of the plot)

> Q. Which geometric layer should be used to create scatter plots in ggplot2? **The geom_point() layer should be used to create scatter plots**

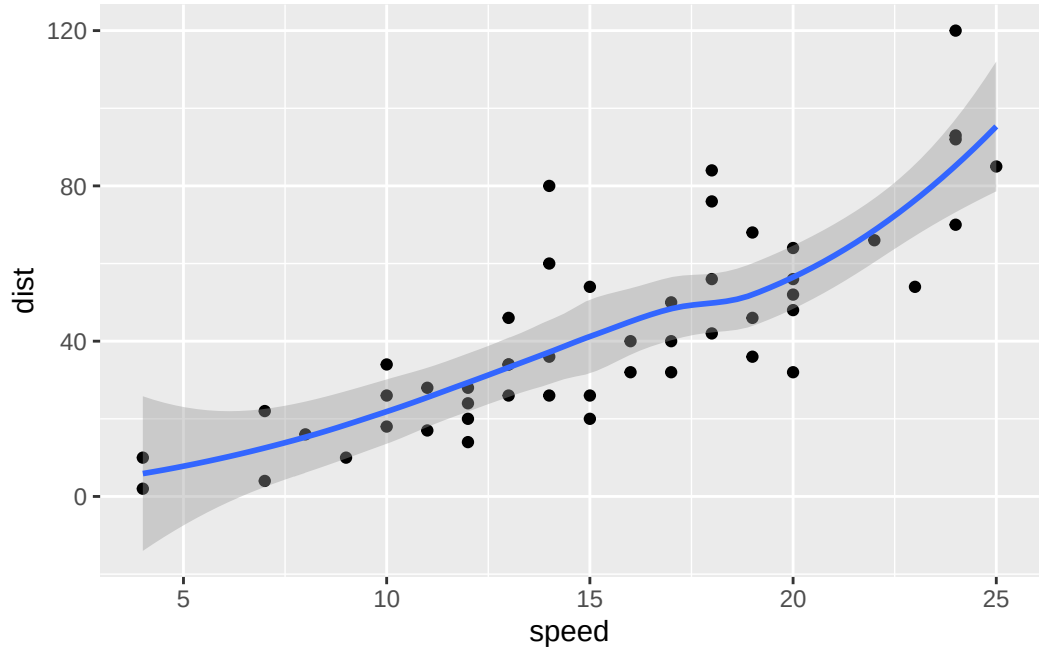
```
ggplot(cars) +
  aes(x=speed, y = dist) +
  geom_point()
```



Q. In your own RStudio can you add a trend line layer to help show the relationship between the plot variables with the geom_smooth() function?

```
ggplot(cars) +
  aes(x=speed, y = dist) +
  geom_point() +
  geom_smooth()
```

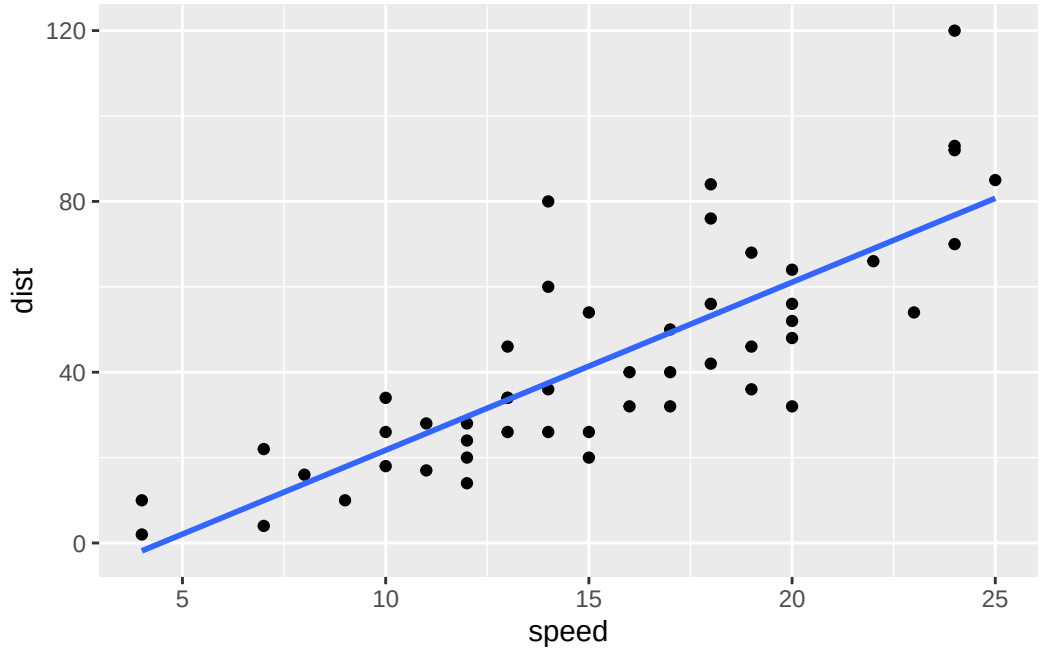
```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Q. Argue with geom_smooth() to add a straight line from a linear model without the shaded standard error region?

```
ggplot(cars) +
  aes(x=speed, y = dist) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)
```

```
`geom_smooth()` using formula = 'y ~ x'
```



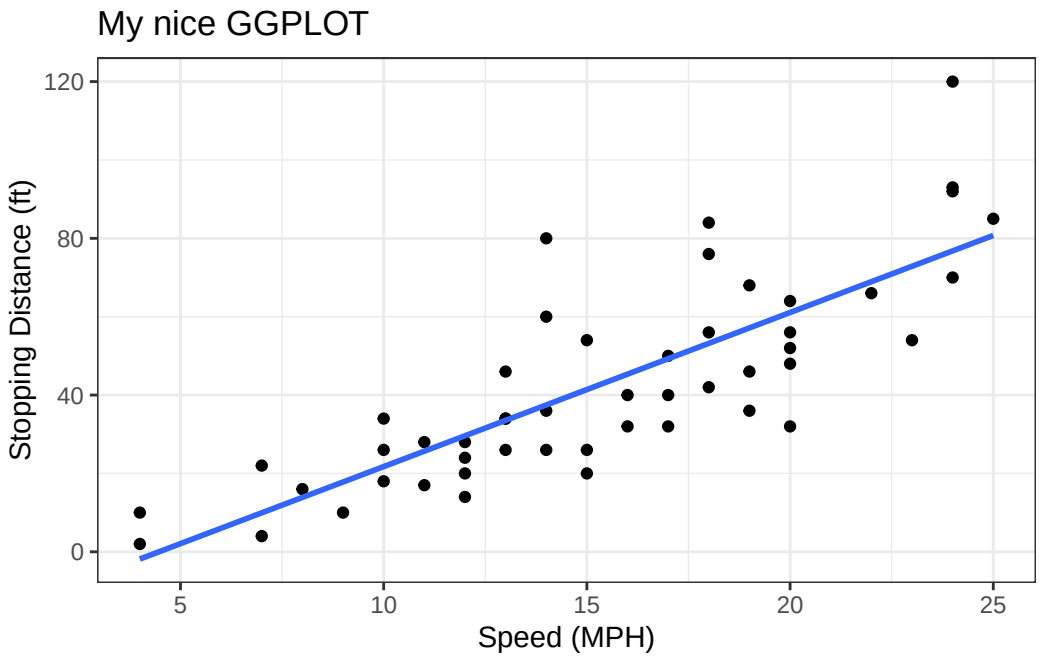
Q. Can you finish this plot by adding various label annotations with the labs() function and changing the plot look to a more conservative "black & white" theme by adding the theme_bw() function:

I can add more layers to build up more complicated plots.

```
p <- ggplot(cars) +
  aes(x=speed, y = dist) +
  geom_point() +
  geom_smooth(method ="lm", se=FALSE)
```

```
p + labs(title="My nice GGPLOT", x = "Speed (MPH)", y = "Stopping Distance (ft)") + theme_bw
```

```
`geom_smooth()` using formula = 'y ~ x'
```



A RNAseq plot with more aes() values

```
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expression.txt"
genes <- read.delim(url)
head(genes)
```

```
        Gene Condition1 Condition2      State
1      A4GNT -3.6808610 -3.4401355 unchanging
2       AAAS  4.5479580  4.3864126 unchanging
3      AASDH  3.7190695  3.4787276 unchanging
4       AATF  5.0784720  5.0151916 unchanging
5       AATK  0.4711421  0.5598642 unchanging
6 AB015752.4 -3.6808610 -3.5921390 unchanging
```

Q. Use the nrow() function to find out how many genes are in this dataset. What is your answer?

There are 5196 genes

```
nrow(genes)
```

```
[1] 5196
```

Q. Use the colnames() function and the ncol() function on the genes data frame to find out what the column names are (we will need these later) and how many columns there are. How many columns did you find?

There are 4 columns: Gene, Condition1, Condition2, and State.

```
colnames(genes)
```

```
[1] "Gene"       "Condition1" "Condition2" "State"
```

```
ncol(genes)
```

```
[1] 4
```

Q. Use the table() function on the State column of this data.frame to find out how many 'up' regulated genes there are. What is your answer?

There are 127 upregrulated genes

```
table(genes$State)
```

```
      down unchanging         up
        72       4997        127
```

Q. Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset?

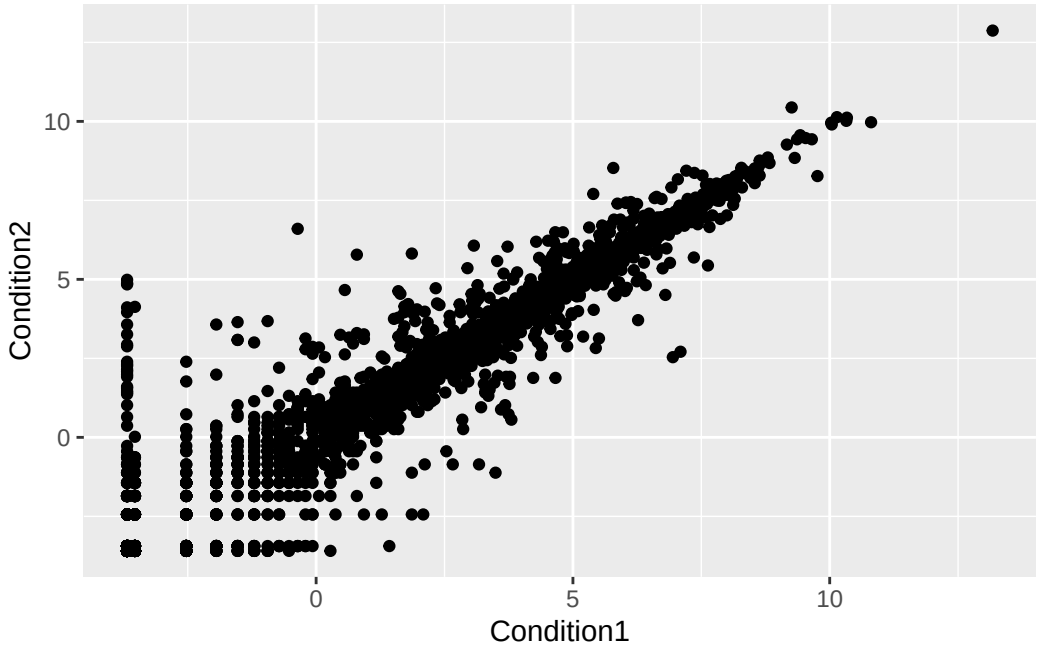
0.024 genes of the total dataset are upregulated

```
signif(table(genes$State) / nrow(genes),2)
```

```
      down unchanging         up
     0.014      0.960     0.024
```

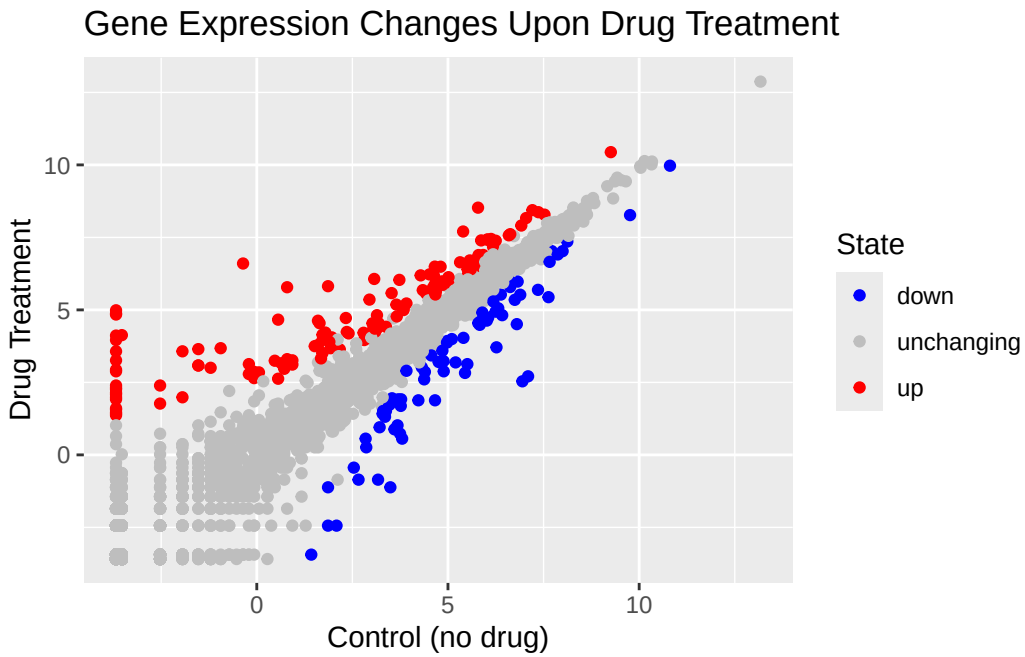
Q. Complete the code below to produce the following plot ggplot*()* + ***aes(x=Condition1, y=***) _____

```
ggplot(genes) +
aes(x=Condition1, y=Condition2) +
geom_point()
```



Q. Nice, now add some plot annotations to the p object with the labs() function so your plot looks like the following:

```
ggplot(genes) +
aes(x=Condition1, y=Condition2, col=State) +
geom_point() +
  scale_colour_manual( values=c("blue","gray","red") ) +
  labs(title = "Gene Expression Changes Upon Drug Treatment", x="Control (no drug)", y = "Dru
```



##Going further section

```
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.ts
```

```
gapminder <- read.delim(url)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
gapminder_2007 <- gapminder %>% filter(year==2007)
```

What years does the dataset cover?

```
unique(gapminder$year)
```

```
 [1] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 2002 2007
```

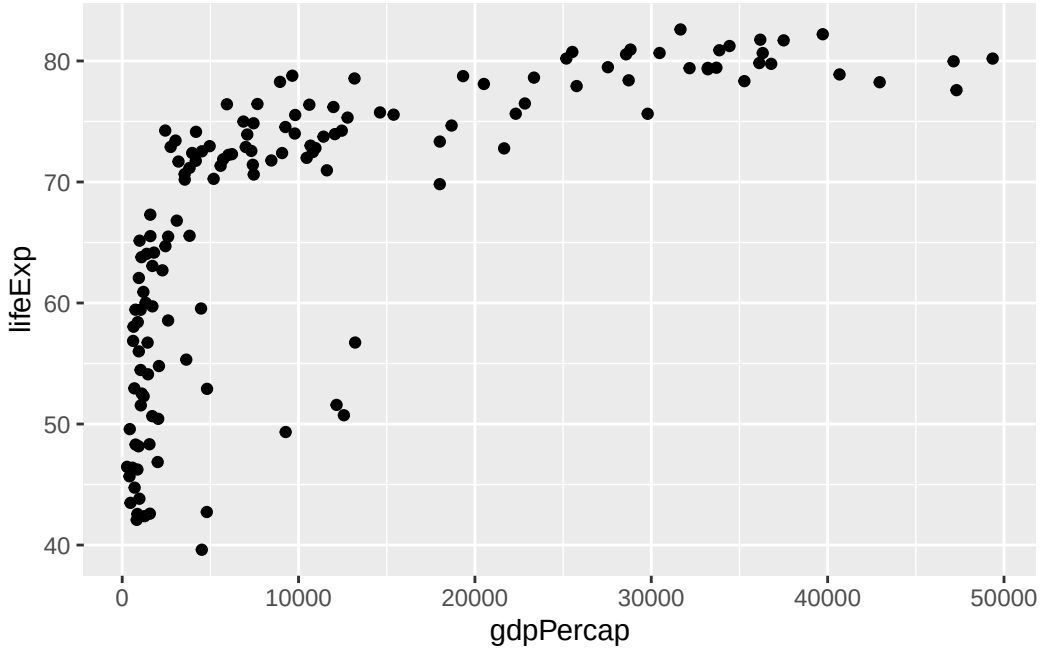
Key functions that will be useful in R include:

nrow(), ncol(), length(), unique(), table()

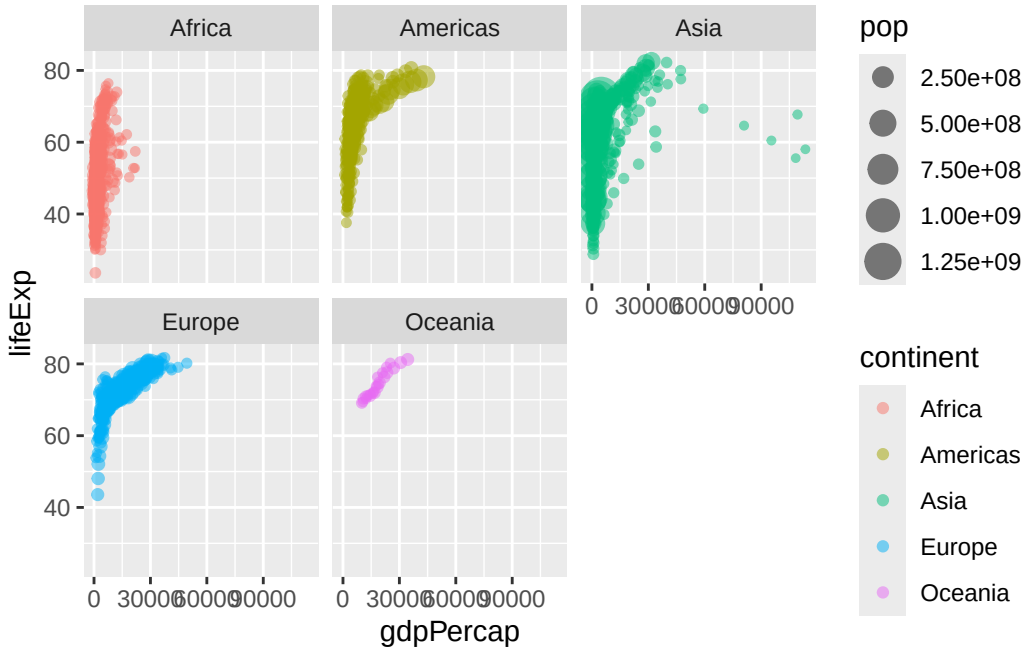
Let's consider the gapminder_2007 dataset which contains the variables GDP per capita gdpPercap and life expectancy lifeExp for 142 countries in the year 2007

> Q. Complete the code below to produce a first basic scater plot of this gapminder_2007 dataset: ggplot(gapminder_2007) + aes(x=, ***y=***) + ____

```
ggplot(gapminder_2007) +
  aes(x= gdpPercap, y=lifeExp) +
  geom_point()
```




```
ggplot(gapminder) +
  aes(x= gdpPercap, y=lifeExp, col=continent, size = pop) +
  geom_point(alpha = 0.5) + facet_wrap(~continent)
```

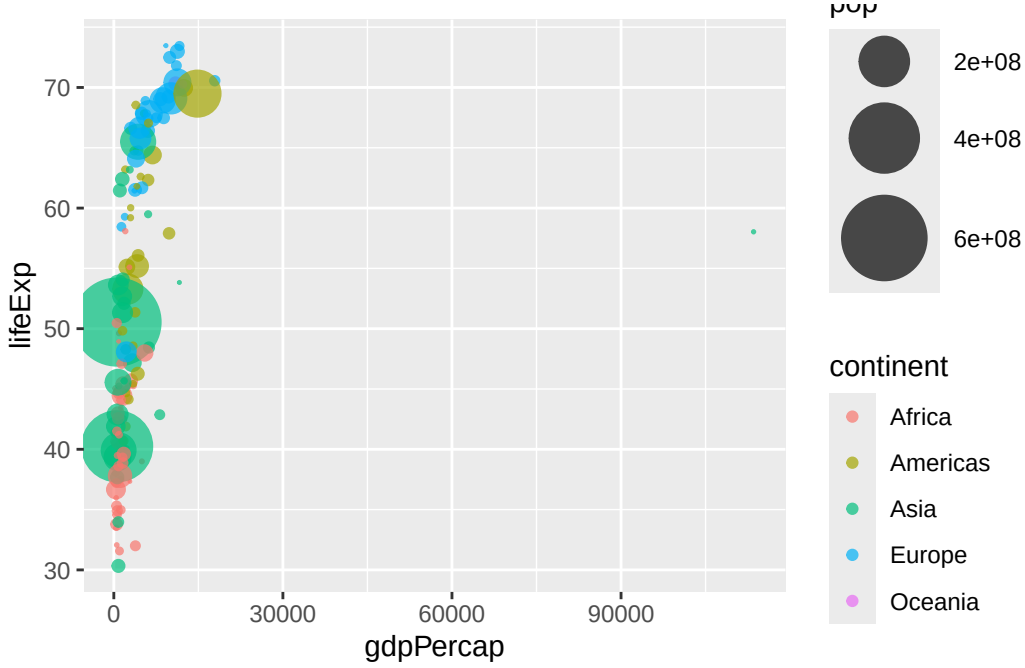


Q. Can you adapt the code you have learned thus far to reproduce our gapminder scatter plot for the year 1957? What do you notice about this plot is it easy to compare with the one for 2007?

Steps to produce your 1957 plot should include:

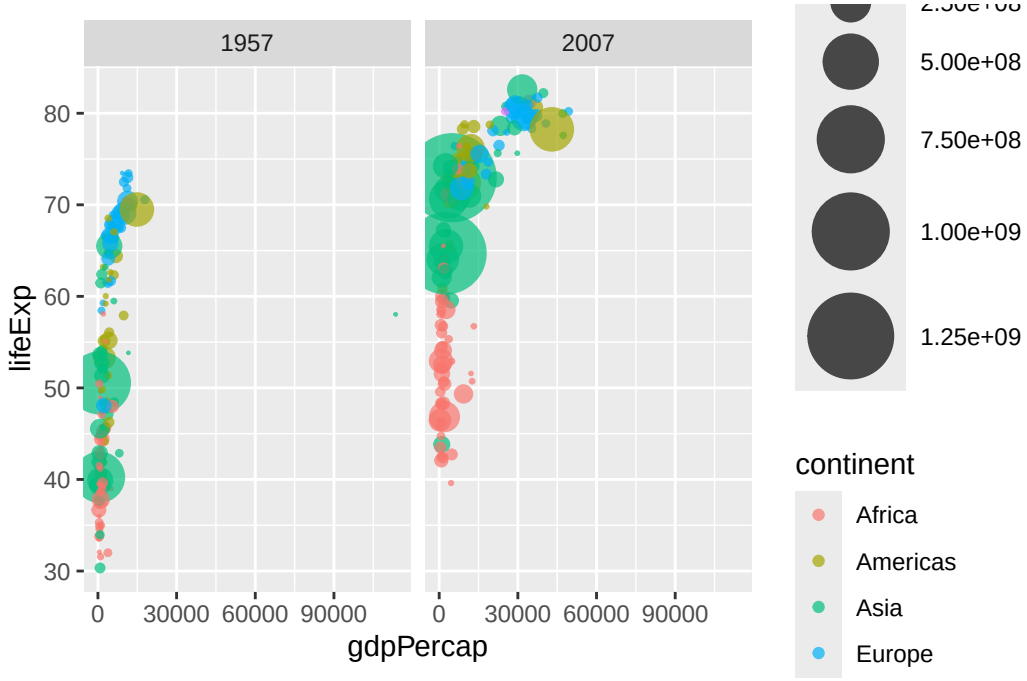
Use dplyr to filter the gapmider dataset to include only the year 1957 (check above for how we did this for 2007). Save your result as gapminder_1957. Use the ggplot() function and specify the gapminder_1957 dataset as input Add a geom_point() layer to the plot and create a scatter plot showing the GDP per capita gdpPercap on the x-axis and the life expectancy lifeExp on the y-axis Use the color aesthetic to indicate each continent by a different color Use the size aesthetic to adjust the point size by the population pop Use scale_size_area() so that the point sizes reflect the actual population differences and set the max_size of each point to 15 -Set the opacity/transparency of each point to 70% using the alpha=0.7 parameter

```
gapminder_1957 <- gapminder %>% filter(year==1957)
ggplot(gapminder_1957) +
  aes(x= gdpPercap, y=lifeExp, color = continent, size = pop) +
  geom_point(alpha = 0.7) +
  scale_size_area(max_size = 15)
```



Q. Do the same steps above but include 1957 and 2007 in your input dataset for ggplot(). You should now include the layer facet_wrap(~year) to produce the following plot:

```
gapminder_1957_2007 <- gapminder %>% filter(year==1957 | year ==2007)
ggplot(gapminder_1957_2007) +
  aes(x= gdpPercap, y=lifeExp, color = continent, size = pop) +
  geom_point(alpha = 0.7) +
  scale_size_area(max_size = 15) +
  facet_wrap(~year)
```



Bar charts

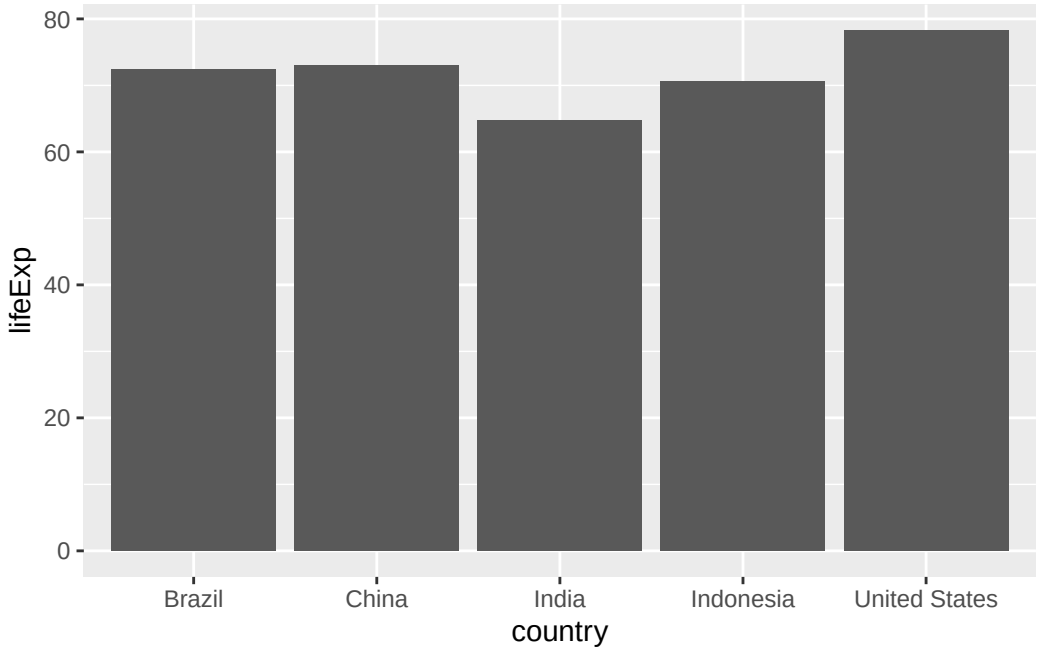
```
gapminder_top5 <- gapminder %>%
  filter(year==2007) %>%
  arrange(desc(pop)) %>%
  top_n(5, pop)

gapminder_top5
```

```
        country continent year lifeExp        pop gdpPercap
1         China      Asia 2007  72.961 1318683096  4959.115
2         India      Asia 2007  64.698 1110396331  2452.210
3 United States  Americas 2007  78.242  301139947 42951.653
4     Indonesia      Asia 2007  70.650  223547000  3540.652
5        Brazil  Americas 2007  72.390  190010647  9065.801
```

Q Create a bar chart showing the life expectancy of the five biggest countries by population in 2007.

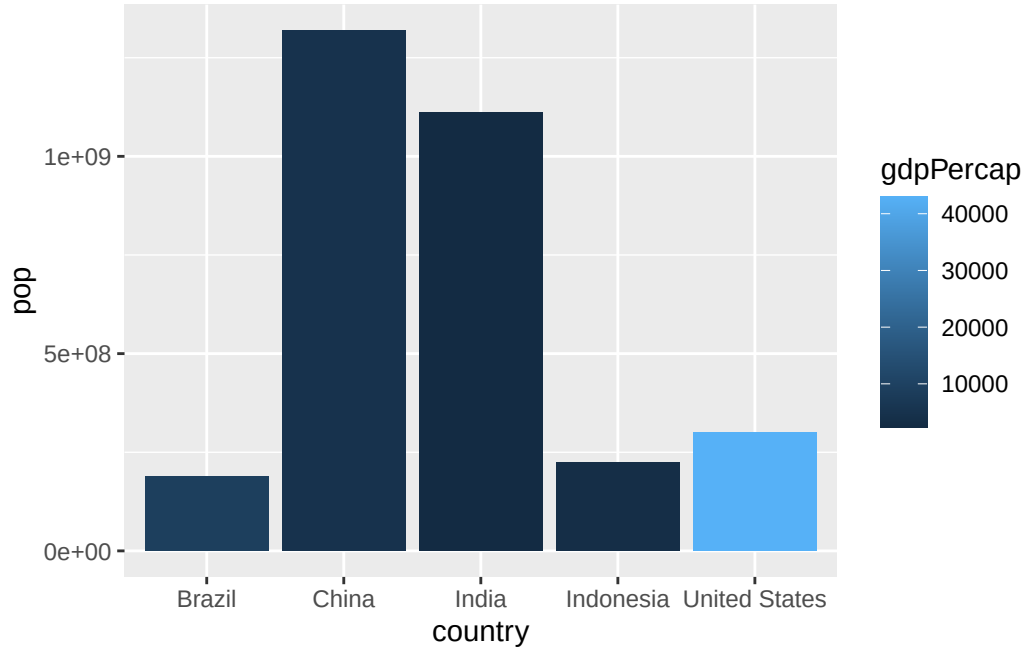
```
ggplot(gapminder_top5) +
  aes(x = country, y =lifeExp) +
  geom_col()
```



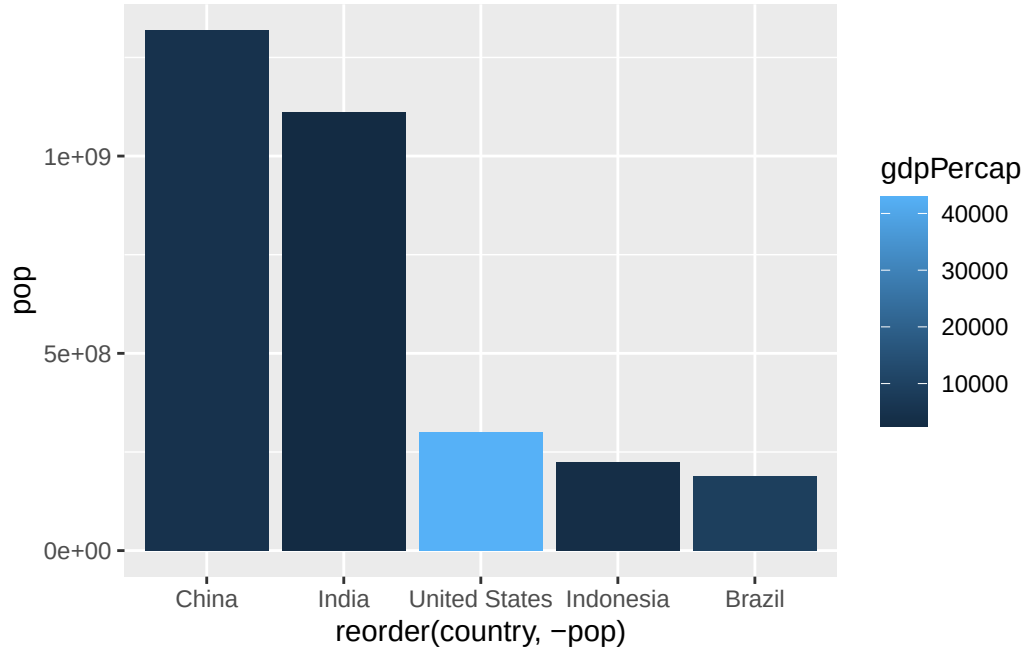
Q. Plot population size by country. Create a bar chart showing the population (in millions) of the five biggest countries by population in 2007.

Use the ggplot() function and specify the gapminder_top5 dataset as input Add a geom_col() layer to the plot Plot one bar for each country (x aesthetic) Use population pop as bar height (y aesthetic) Use the GDP per capita gdpPercap as fill aesthetic

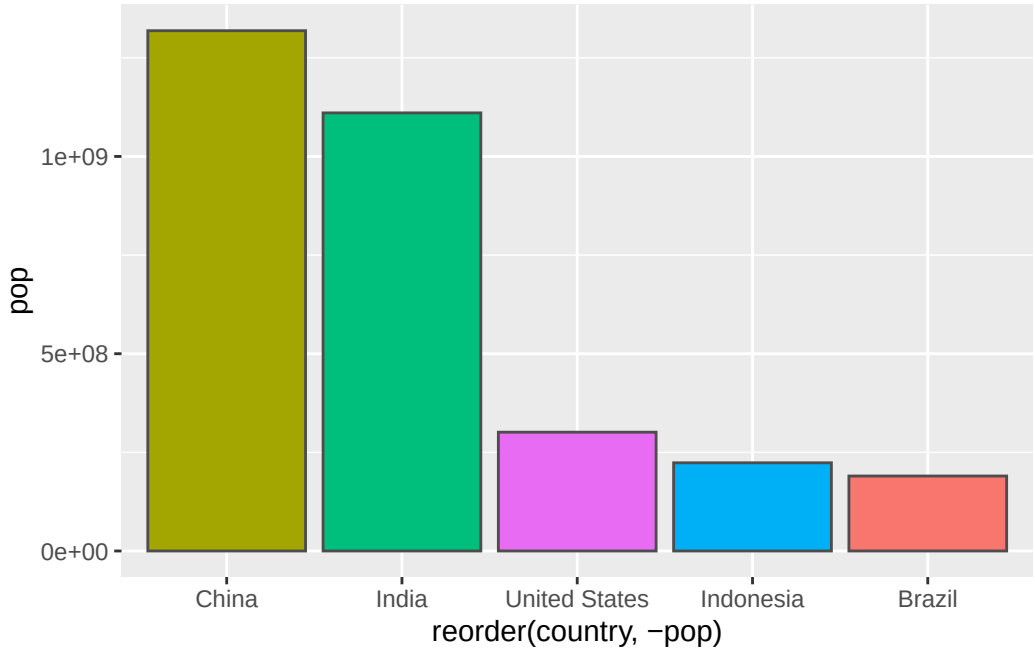
```
ggplot(gapminder_top5) +
  aes(x=country, y=pop, fill = gdpPercap) +
  geom_col()
```




```
ggplot(gapminder_top5) +
  aes(x=reorder(country, -pop), y=pop, fill=gdpPercap) +
  geom_col()
```



```
ggplot(gapminder_top5) +
  aes(x=reorder(country, -pop), y=pop, fill=country) +
  geom_col(col="gray30") +
  guides(fill="none")
```



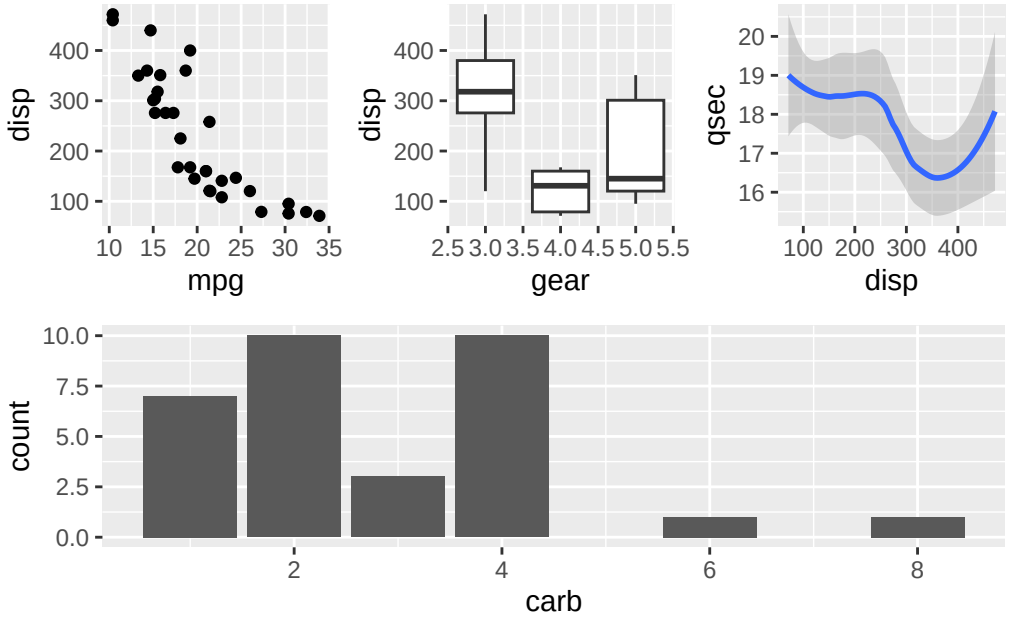
##Combining plots

```
library(patchwork)

# Setup some example plots
p1 <- ggplot(mtcars) + geom_point(aes(mpg, disp))
p2 <- ggplot(mtcars) + geom_boxplot(aes(gear, disp, group = gear))
p3 <- ggplot(mtcars) + geom_smooth(aes(disp, qsec))
p4 <- ggplot(mtcars) + geom_bar(aes(carb))

# Use patchwork to combine them here:
(p1 | p2 | p3) /
      p4
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```


disp
400
300
200
100
10 15 20 25 30 35
mpg
disp
400
300
200
100
2.5 3.0 3.5 4.0 4.5 5.0 5.5
gear
qsec
20
19
18
17
16
100 200 300 400
disp
count
10.0
7.5
5.0
2.5
0.0
2 4 6 8
carb